

Kierunek: Informatyka Algorytmiczna (INA)

Specjalność: -

PRACA DYPLOMOWA
INŻYNIERSKA

**Implementacja programu GRAPHPLAN do
planowania akcji z wykorzystaniem
programowania ograniczeń**

Radosław Wojtczak

Opiekun pracy
dr Przemysław Kobyłański

Słowa kluczowe:

Planowanie

Grafy

Programowanie ograniczeń

Sztuczna inteligencja

Streszczenie

Obiektem badań poniższej pracy jest metodologia planowania o nazwie "Graphplan", której esencją jest wykorzystanie własności grafów w trakcie ustalania optymalnego planu transformacji stanu początkowego w stan końcowy w ustalonej przestrzeni przy wykorzystaniu wcześniej zdefiniowanych operatorów. Ów praca składa się z formalnego opisu przytoczonego algorytmu, przedstawienia przykładów zastosowania, implementacji, której wynikiem jest graf, przedstawiający optymalny plan wykonywanych operacji, omówienie opcjonalnych rozszerzeń, które w zależności od sytuacji mogą wpłynąć na efektywność algorytmu, oraz przeprowadzonych testów, których zadaniem jest wskazanie mocnych, jak i słabych stron przedmiotu badań.

Abstract

Tutaj treść streszczenia po angielsku.

Spis treści

Spis rysunków	II
Spis tabel	III
Wstęp	1
1 Analiza problemu	3
2 Projekt systemu	5
2.1 Grupy użytkowników i założenia	5
2.2 Przypadki użycia i scenariusze	5
2.3 Diagramy klas	5
2.4 Diagramy aktywności	5
2.5 Diagramy sekwencji	5
2.6 Diagramy stanów	6
2.7 Projekt bazy danych	6
2.8 Opis protokołów	6
2.9 Opis algorytmów	7
3 Implementacja systemu	9
3.1 Opis technologii	9
3.2 Omówienie kodów źródłowych	9
4 Instalacja i wdrożenie	11
Podsumowanie	13
Bibliografia	15
A Zawartość płyty CD	17

Spis rysunków

2.1	Diagram aktywności związany z procesem rejestracji dokumentu.	6
-----	---	---

Spis tabel



Wstęp

Praca dyplomowa inżynierska jest dokumentem opisującym zrealizowany system techniczny. Praca powinna być napisana poprawnym językiem odzwierciedlającym aspekty techniczne (informatyczne) omawianego zagadnienia. Praca powinna być napisana w trybie bezosobowym (w szczególności należy unikać trybu pierwszej osoby liczby pojedynczej i mnogiej). Zdania opisujące konstrukcję systemu informatycznego powinny być tworzone w stronie biernej. W poniższym dokumencie przykłady sformułowań oznaczono kolorem niebieskim. W opisie elementów systemu, komponentów, elementów kodów źródłowych, nazw plików, wejść i wyjść konsoli należy stosować czcionkę stałej szerokości, np: `zmienna wynik przyjmuje wartość zwracaną przez funkcję dodaj(a,b)`, dla argumentów `a` oraz `b` przekazywanych ...

Układ poniższego dokumentu przedstawia wymaganą strukturę pracy, z rozdziałami zawierającymi analizę zagadnienia, opis projektu systemu oraz implementację (dobór podrozdziałów jest przykładowy i należy go dostosować do własnej tematyki pracy).

Wstęp pracy (nie numerowany) powinien składać się z czterech części (które nie są wydzielane jako osobne podrozdziały): zakresu pracy, celu, analizy i porównania istniejących rozwiązań oraz przeglądu literatury, oraz opisu zawartości pracy.

Każdy rozdział powinien rozpoczynać się od akapitu wprowadzającego, w którym zostaje w skrócie omówiona zawartość tego rozdziału.

Praca swoim zakresem obejmuje wielowarstwowe rozproszone systemy informatyczne typu „B2B” wspierające wymianę danych pomiędzy przedsiębiorstwami. Systemy tego typu, konstruowane dla dużych korporacji, charakteryzują się złożoną strukturą poziomą i pionową, w której dokumenty ...

Celem pracy jest zaprojektowanie i oprogramowanie aplikacji o następujących założeniach funkcjonalnych:

- wspieranie zarządzania obiegiem dokumentów wewnątrz korporacji z uwzględnieniem ... ,
- wspieranie zarządzania zasobami ludzkimi z uwzględnieniem modułów kadrowych oraz ... ,
- gotowość do uzyskania certyfikatu ISO ... ,
-

Istnieje szereg aplikacji o zbliżonej funkcjonalności: ... , przy czym

Praca składa się z czterech rozdziałów. W rozdziale pierwszym omówiono strukturę przedsiębiorstwa ... , scharakteryzowano grupy użytkowników oraz przedstawiono procedury związane z obiegiem dokumentów. Szczegółowo opisano mechanizmy Przedstawiono uwarunkowania prawne udostępniania informacji ... , w ramach

W rozdziale drugim przedstawiono szczegółowy projekt systemu w notacji UML. Wykorzystano diagramy Opisano w pseudokodzie i omówiono algorytmy generowania

W rozdziale trzecim opisano technologie implementacji projektu: wybrany język programowania, biblioteki, system zarządzania bazą danych, itp. Przedstawiono dokumentację techniczną kodów źródłowych interfejsów poszczególnych modułów systemu. Przedstawiono sygnatury metod publicznych oraz

W rozdziale czwartym przedstawiono sposób instalacji i wdrożenia systemu w środowisku docelowym.

Końcowy rozdział stanowi podsumowanie uzyskanych wyników.



Rozdział 1

Analiza problemu



Rozdział 2

Projekt systemu

W tym rozdziale przedstawiono szczegółowy projekt systemu w notacji UML uwzględniający wymagania funkcjonalne opisane w rozdziale 1. Do opisu relacji pomiędzy składowymi systemu wykorzystano diagramy Przedstawiono w pseudokodzie i omówiono algorytmy generowania

2.1 Grupy użytkowników i założenia

Architektura systemu ...jest wielowarstwowa i rozproszona, przy czym Podsystem ...jest systemem zbiorczym dla danych ... wysyłanych do serwera

Taka architektura jest zgodna z wzorcem projektowym MVC¹ (ang. Model-View-Controller). Przetwarzanie danych odbywa się

2.2 Przypadki użycia i scenariusze

W tej sekcji należy przedstawić przypadki użycia oraz odpowiadające im scenariusze dla poszczególnych grup użytkowników

2.3 Diagramy klas

W tej sekcji należy przedstawić diagramy klas dla odpowiednich elementów systemu zidentyfikowane na podstawie wcześniejszych rozważań

2.4 Diagramy aktywności

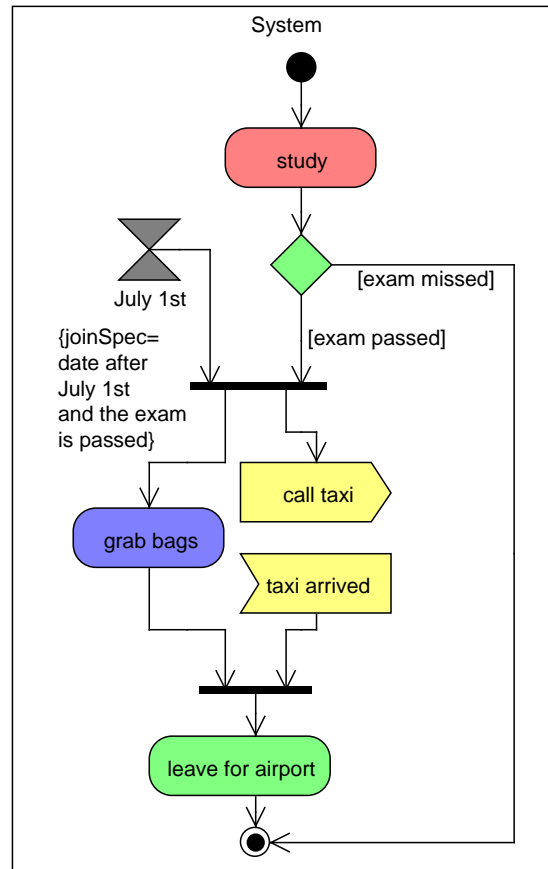
W tej sekcji należy przedstawić diagramy aktywności dla elementów systemu i odpowiednich procesów wynikające z wcześniejszej analizy.

W niniejszym rozdziale przedstawiono diagramy aktywności Diagram na rysunku 2.1 przedstawia

2.5 Diagramy sekwencji

W tej sekcji należy przedstawić diagramy sekwencji dla obiektów systemu zidentyfikowanych na podstawie wcześniejszych rozważań. Należy wykorzystać nazewnictwo wprowadzone w poprzednich rozdziałach, w szczególności odpowiadające definicjom wprowadzonych klas.

¹Należy odnieść się do wykorzystywanych wzorców projektowych



Rysunek 2.1: Diagram aktywności związany z procesem rejestracji dokumentu.

2.6 Diagramy stanów

W tej sekcji należy przedstawić diagramy stanów w których może znaleźć się system. Diagramy te są szczególnie istotne przy projektowaniu systemów czasu rzeczywistego.

2.7 Projekt bazy danych

W tej sekcji należy przedstawić projekt bazy danych. Należy omówić wycinek rzeczywistości i odpowiadające mu zidentyfikowane elementy systemu, których wartości będą podlegać utrwalaniu. Należy przedyskutować wybór typów danych dla atrybutów poszczególnych obiektów. Należy uzasadnić wybór platformy DBMS. Dla relacyjnych baz danych należy przedyskutować jej normalizację.

2.8 Opis protokołów

W tej sekcji należy omówić protokoły wykorzystywane przez komponenty systemu. Omówić formaty komunikatów i zilustrować je przykładami.

2.9 Opis algorytmów

W tej sekcji należy wymienić i przedyskutować algorytmy wykorzystywane w systemie. Algorytmy należy przedstawić w pseudokodzie (wykorzystać pakiet `algorithm2e`). Omówienia poszczególnych kroków algorytmów powinny zawierać odwołania do odpowiednich linii pseudokodu. Dla zaproponowanych autorskich algorytmów należy przeprowadzić analizę ich złożoności czasowej i pamięciowej.

Algorytm bąblowania jest przedstawiony w Pseudokodzie 2.1.

Pseudokod 2.1: Wyporność przez bąblowanie

Input: Zbiór bąbli B

Output: Wyporność W

```
1 foreach  $b \in B$  do
2   Process( $b$ );
3   for  $i \leftarrow 1$  to  $|B|$  do
4     if Calculate( $EW(i, b)$ )  $\leq 0$  then
5        $b \leftarrow 2 * b$ ;
6 while  $B \neq \emptyset$  do
7   for  $j \leftarrow 1$  to  $|B|$  do
8     if Calculate( $FT(j, \hat{b})$ )  $\leq 0$  then
9        $w \leftarrow 2 * \hat{b}$ ;
10       $W \leftarrow W \cup \{w\}$ ;
11       $B \leftarrow B \setminus \{b\}$ ;
```



Rozdział 3

Implementacja systemu

3.1 Opis technologii

Należy tutaj zamieścić krótki opis (z referencjami) do technologii użytych przy implementacji systemu.

Do implementacji systemu użyto języka JAVA w wersji ..., szczegółowy opis można znaleźć w [1]. Interfejs zaprojektowano w oparciu o HTML5 i CSS3 [?].

3.2 Omówienie kodów źródłowych

Kod źródłowy 3.1 przedstawia opisy poszczególnych metod interfejsu: WSPodmiotRejestracjaIF. Kompletne kody źródłowe znajdują się na płycie CD dołączonej do niniejszej pracy w katalogu Kody (patrz Dodatek A).

Kod źródłowy 3.1: Interfejs usługi Web Service: WSPodmiotRejestracjaIF.

```
package erejestracja.podmiot;
import java.rmi.RemoteException;
// Interfejs web serwisu dotyczącego obsługi podmiotów i rejestracji.
public interface WSPodmiotRejestracjaIF extends java.rmi.Remote{
// Pokazuje informacje o danym podmiocie.
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.
// return: Podmiot – obiekt transportowy: informacje o danym podmiocie.
public Podmiot pokazPodmiot(long nrPeselRegon) throws RemoteException;
// Dodaje nowy podmiot.
// parametr: nowyPodmiot – obiekt transportowy: informacje o nowym podmiocie.
// return: true – jeśli podmiot dodano, false – jeśli nie dodano.
public boolean dodajPodmiot(Podmiot nowyPodmiot) throws RemoteException;
// Usuwa dany podmiot.
// parametr: nrPeselRegon – numer PESEL osoby fizycznej lub numer REGON firmy.
// return: true – jeśli podmiot usunięto, false – jeśli nie usunięto.
public boolean usunPodmiot(long nrPeselRegon) throws RemoteException;
// Modyfikuje dany podmiot.
// parametr: podmiot – obiekt transportowy: informacje o modyfikowanym podmiocie.
// return: true – jeśli podmiot zmodyfikowano, false – jeśli nie zmodyfikowano.
public boolean modyfikujPodmiot(Podmiot podmiot) throws RemoteException;
// Pokazuje zarejestrowane podmioty na dany dowód rejestracyjny.
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.
// return: PodmiotRejestracja[] – tablica obiektów transportowych: informacje o
// wszystkich zarejestrowanych podmiotach.
public PodmiotRejestracja[] pokazZarejestrowanePodmioty(
String nrDowoduRejestracyjnego) throws RemoteException;
// Nowa rejestracja podmiotu na dany dowód rejestracyjny.
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.
```



```
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.  
// parametr: czyWlasciciel – czy dany podmiot jest właścicielem pojazdu.  
// return: true – jeśli zarejestrowano podmiot, false – jeśli nie zarejestrowano.  
public boolean zarejestrujNowyPodmiot(String nrDowoduRejestracyjnego,  
long nrPeselRegon, boolean czyWlasciciel) throws RemoteException;  
// Usuwa wiązanie pomiędzy danym podmiotem, a dowodem rejestracyjnym.  
// parametr: nrDowoduRejestracyjnego – numer dowodu rejestracyjnego.  
// parametr: nrPeselRegon – numer PESEL podmiotu lub numer REGON firmy.  
// return: true – jeśli podmiot wyrejestrowano, false – jeśli nie wyrejestrowano.  
public boolean wyrejestrujPodmiot(String nrDowoduRejestracyjnego,  
long nrPeselRegon) throws RemoteException;
```

Kod źródłowy [3.2](#) przedstawia procedurę przetwarzającą żądanie. Hasz utrwalany `%granulacja` wykorzystywany jest do komunikacji międzyprocesowej.

Kod źródłowy 3.2: Przetwarzanie zadania - procedura `process_req()`.

```

sub process_req(){
my($r) = @_;
$wyn = "";
if ($r=~get/i) {
    @request = split("_",$r);
    $zad = $request[0];
    $ts1 = $request[1];
    $ts2 = $request[2];
    @date1 = split(/\D/,$ts1);
    @date2 = split(/\D/,$ts2);
    print "odebralem:_$r";
    $wyn = $wyn."zadanie:_$zad\n";
    $wyn = $wyn."czas_od:_".$date1[0]"_"$date1[1]"_"$date1[2]"_"$date1[3]" ":" "$date1[4]" ":" "$date1[5]" ":" "$date1[6]" ":" "$date1[7]";
    $wyn = $wyn."czas_do:_".$date2[0]"_"$date2[1]"_"$date2[2]"_"$date2[3]" ":" "$date2[4]" ":" "$date2[5]" ":" "$date2[6]" ":" "$date2[7]";
    $wyn = $wyn.&sym_sens($ts1,$ts2);
    return $wyn;
}
if ($r=~set gt/i) {
    @request = split("_",$r);
    $zad = $request[0];
    $ts1 = $request[1];
    $ts2 = $request[2];
    $gt = $request[2];
    dbmopen(%granulacja,"granulacja_baza",0644);
    $granulacja{"gt"}=$gt;
    dbmclose(%granulacja);
    $wyn = "\'GT\'_zmienione_na:_$gt";
}
}

```

Rozdział 4

Instalacja i wdrożenie

W tym rozdziale należy omówić zawartość pakietu instalacyjnego oraz założenia co do środowiska, w którym realizowany system będzie instalowany. Należy przedstawić procedurę instalacji i wdrożenia systemu. Czynności instalacyjne powinny być szczegółowo rozpisane na kroki. Procedura wdrożenia powinna obejmować konfigurację platformy sprzętowej, OS (np. konfigurację niezbędnych sterowników) oraz konfigurację wdrażanego systemu, m.in. tworzenia niezbędnych kont użytkowników. Procedura instalacji powinna prowadzić od stanu, w którym nie są zainstalowane żadne składniki systemu, do stanu w którym system jest gotowy do pracy i oczekuje na akcje typowego użytkownika.



Podsumowanie

W podsumowaniu należy określić stan zakończonych prac projektowych i implementacyjnych. Zaznaczyć, które z zakładanych funkcjonalności systemu udało się zrealizować. Omówić aspekty pielęgnacji systemu w środowisku wdrożeniowym. Wskazać dalsze możliwe kierunki rozwoju systemu, np. dodawanie nowych komponentów realizujących nowe funkcje.

W podsumowaniu należy podkreślić nowatorskie rozwiązania zastosowane w projekcie i implementacji (niebanalne algorytmy, nowe technologie, itp.).



Bibliografia

- [1] I. Bratko. *Prolog Programming for Artificial Intelligence*. Pearson, 2012.



Załącznik A

Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.

